



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/817,466	04/02/2004	Jun Xu	CIS0213US	3179
33031 7590 01/06/2009 CAMPBELL STEPHENSON LLP 11401 CENTURY OAKS TERRACE BLDG. H, SUITE 250 AUSTIN, TX 78758				
EXAMINER				
YIGDALL, MICHAEL J				
ART UNIT		PAPER NUMBER		
2192				
MAIL DATE		DELIVERY MODE		
01/06/2009		PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/817,466

Applicant(s)

XU ET AL.

Examiner

Michael J. Yigdall

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 12 December 2008.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-39 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-39 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SE/US)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on December 12, 2008 has been entered. Claims 1-39 are pending.

Response to Arguments

2. Applicant's arguments have been considered but are moot in view of the new ground(s) of rejection set forth below. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action.

Claim Rejections under 35 U.S.C. § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. § 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 34-39 are rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter that Applicant regards as the invention.

With respect to claim 34 (currently amended), there is insufficient antecedent basis for "the program under test" and "the means for debugging" such as recited in the claim. In view of Applicant's amendment, the examiner's presumption is that the first instance of "the program

under test” should be replaced with --a program under test-- and that “the means for debugging” should be replaced with --the debugger program--.

With respect to claims 35-39 (currently amended), the claims are dependent upon claim 34 and therefore are indefinite for the same reason(s) as noted above. Furthermore, claims 35-38 each recite “the means for debugging” such as noted above.

Claim Rejections under 35 U.S.C. § 103

5. The following is a quotation of 35 U.S.C. § 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claims 1-39 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,903,718 to Marik (already of record, “Marik”) in view of U.S. Pub. No. 2003/0074650 to Akgul et al. (now made of record, “Akgul”) and in view of U.S. Patent No. 7,296,187 to Fritz et al. (already of record, “Fritz”).

With respect to claim 1 (currently amended), Marik teaches a method comprising:
providing a debugger agent (see, for example, column 9, lines 16-18, which shows a debugger agent in the form of PC Host Debugger Application), wherein

the debugger agent is configured to select a debugger program suitable for a device under test (see, for example, column 2, lines 23-31, which shows a debugger

program suitable for a device under test, and note that the debugger agent is implicitly configured to select the debugger program).

Marik does not explicitly describe that the debugger agent is further configured to select the debugger program from a plurality of debugger programs.

Nonetheless, in an analogous art, Akgul teaches a method and system for debugging a device under test (see, for example, FIG. 1 and paragraph [0017]). The system selects a debugger module from a plurality of debugger modules and loads the debugger module into the device under test (see, for example, paragraph [0042]). Akgul describes that loading only the necessary debugger modules conserves storage space on the device under test (see, for example, paragraph [0018]).

One of ordinary skill in the art could, with predictable results, incorporate such selection and loading into the teachings of Marik. Thus, it would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the teachings of Marik such that the debugger agent is further configured to select the debugger program from a plurality of debugger programs. As Akgul suggests, such an implementation would conserve storage space on the device under test.

Marik in view of Akgul further teaches or suggests:

the device under test is configured to execute a program under test (see, for example, column 2, lines 23-27, which shows that the device under test is configured to execute a program under test);

causing the debugger agent to load the debugger program into the device under test (see, for example, column 8, lines 23-25, which shows the debugger agent loading the debugger

program into the device under test, and see, for example, column 12, lines 13-18, which further shows that debugger routines are downloaded into the device under test);

sending a plurality of test commands to the device under test (see, for example, column 8, line 53 to column 9, line 1, which shows sending a plurality of test commands in the form of D-packets to the device under test).

Marik does not explicitly describe that the plurality of test commands are sent to the device under test according to a test script.

Nonetheless, in an analogous art, Fritz teaches a method and system for debugging a device under test (see, for example, column 4, lines 13-21). Specifically, Fritz teaches sending a plurality of test commands to the device under test according to a test script (see, for example, column 4, lines 27-40). The test script enables the user to define complex test actions (see, for example, column 6, lines 29-55).

One of ordinary skill in the art could, with predictable results, incorporate such a test script into the teachings of Marik. Thus, it would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the teachings of Marik such that the plurality of test commands are sent to the device under test according to a test script. As Fritz suggests, such an implementation would enable the user to define complex test actions.

Marik in view of Akgul and Fritz further teaches or suggests:

activating the debugger program when a watched event occurs during execution of the program under test (see, for example, column 8, lines 34-50, which shows activating the debugger program when a watched event is encountered in the program under test).

With respect to claim 2 (previously presented), the rejection of claim 1 is incorporated, and Marik in view of Akgul and Fritz further teaches or suggests:

directing a debugger command to the debugger program (see, for example, column 8, line 64 to column 9, line 1, which shows sending a command to the debugger program); and

recording information provided by the debugger program according to the debugger command (see, for example, column 9, lines 1-10, which shows recording information from the debugger program according to the command).

With respect to claim 3 (previously presented), the rejection of claim 1 is incorporated, and Marik in view of Akgul and Fritz further teaches or suggests:

pausing execution of the program under test (see, for example, column 8, lines 53-63, which shows breaking or pausing execution of the program under test); and

allowing a user to control the debugger program (see, for example, column 8, line 64 to column 9, line 1, which shows allowing a user to control the debugger program).

With respect to claim 4 (previously presented), the rejection of claim 1 is incorporated, and Marik in view of Akgul and Fritz further teaches or suggests:

invoking the debugger program while specifying the program under test as a target of the debugger program (see, for example, column 14, lines 35-58, which shows invoking the debugger program while implicitly specifying the program under test as the target).

With respect to claim 5 (previously presented), the rejection of claim 1 is incorporated, and Marik in view of Akgul and Fritz further teaches or suggests:

instructing the debugger program to associate itself with a process executing on the device under test, wherein the process corresponds to the program under test (see, for example, column 14, lines 35-58, which shows implicitly associating the debugger with a process that corresponds to the program under test).

With respect to claim 6 (previously presented), the rejection of claim 1 is incorporated, and Marik in view of Akgul and Fritz further teaches or suggests:

sending a command to the debugger program (see, for example, column 15, line 50 to column 16, line 3, which shows sending a command to the debugger program to set a debugpoint in the program under test), wherein the command performs at least one of:

- setting a breakpoint in the program under test;
- setting a watchpoint in the program under test;
- setting a catchpoint in the program under test; and
- setting a tracepoint in the program under test

(see, for example, column 14, lines 35-39, which shows that the debugpoint is at least one of a breakpoint, a tracepoint and a steppoint).

With respect to claim 7 (previously presented), the rejection of claim 1 is incorporated, and Marik in view of Akgul and Fritz further teaches or suggests that the watched event comprises at least one of:

- a processor exception;
- a program under test error;
- reaching a breakpoint in the program under test;

reaching a watchpoint in the program under test;
reaching a catchpoint in the program under test; and
reaching a tracepoint in the program under test
(see, for example, column 16, lines 19-35, which shows that the watched event comprises reaching breakpoint in the program under test).

With respect to claim 8 (previously presented), the rejection of claim 1 is incorporated, and Marik in view of Akgul and Fritz further teaches or suggests:

selecting a platform-specific debugger program corresponding to a processor in the device under test (see, for example, column 7, line 49 to column 8, line 3, which shows that the debugger program is platform-specific and corresponds to a microcontroller in the device under test); and

loading the platform-specific debugger program into the device under test (see, for example, column 2, lines 23-31, which shows that the debugger program is loaded into the device under test).

With respect to claim 9 (previously presented), the rejection of claim 8 is incorporated, and Marik in view of Akgul and Fritz further teaches or suggests:

loading, into the device under test, a symbol file corresponding to the program under test (see, for example, column 10, lines 25-45, which shows downloading the program under test and implicitly a corresponding symbol file to the device under test).

With respect to claim 10 (currently amended), Marik teaches a system comprising:

a memory (see, for example, PC Host 10 in FIG. 1, which implicitly includes a memory);

a processor coupled to the memory (see, for example, PC Host 10 in FIG. 1, which implicitly includes a processor coupled to the memory); and

a debugger agent, wherein at least a portion of the debugger agent is encoded as instructions stored in the memory and executable on the processor (see, for example, column 9, lines 16-18, which shows a debugger agent in the form of PC Host Debugger Application), and wherein the debugger agent is configured to:

select a debugger program suitable for a device under test, wherein the device under test is configured to execute a program under test (see, for example, column 2, lines 23-31, which shows a debugger program suitable for a device under test that is configured to execute a program under test, and note that the debugger agent is implicitly configured to select the debugger program).

Marik does not explicitly describe that the debugger agent is further configured to select the debugger program from a plurality of debugger programs.

Nonetheless, in an analogous art, Akgul teaches a method and system for debugging a device under test (see, for example, FIG. 1 and paragraph [0017]). The system selects a debugger module from a plurality of debugger modules and loads the debugger module into the device under test (see, for example, paragraph [0042]). Akgul describes that loading only the necessary debugger modules conserves storage space on the device under test (see, for example, paragraph [0018]).

One of ordinary skill in the art could, with predictable results, incorporate such selection and loading into the teachings of Marik. Thus, it would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the

teachings of Marik such that the debugger agent is further configured to select the debugger program from a plurality of debugger programs. As Akgul suggests, such an implementation would conserve storage space on the device under test.

Marik in view of Akgul further teaches or suggests that the debugger agent is configured to:

cause the debugger program to be loaded into the device under test (see, for example, column 8, lines 23-25, which shows the debugger agent loading the debugger program into the device under test, and see, for example, column 12, lines 13-18, which further shows that debugger routines are downloaded into the device under test),

send a plurality of test commands to the device under test (see, for example, column 8, line 53 to column 9, line 1, which shows sending a plurality of test commands in the form of D-packets to the device under test).

Marik does not explicitly describe that the plurality of test commands are sent to the device under test according to a test script.

Nonetheless, in an analogous art, Fritz teaches a method and system for debugging a device under test (see, for example, column 4, lines 13-21). Specifically, Fritz teaches sending a plurality of test commands to the device under test according to a test script (see, for example, column 4, lines 27-40). The test script enables the user to define complex test actions (see, for example, column 6, lines 29-55).

One of ordinary skill in the art could, with predictable results, incorporate such a test script into the teachings of Marik. Thus, it would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the teachings of

Marik such that the plurality of test commands are sent to the device under test according to a test script. As Fritz suggests, such an implementation would enable the user to define complex test actions.

Marik in view of Akgul and Fritz further teaches or suggests that the debugger agent is configured to:

activate the debugger program when a watched event occurs during execution of the program under test (see, for example, column 8, lines 34-50, which shows activating the debugger program when a watched event is encountered in the program under test).

With respect to claim 11 (original), the rejection of claim 10 is incorporated, and Marik in view of Fritz further teaches or suggests at least one debugger program stored in at least one of the memory and a storage device accessible by the processor (see, for example, column 8, lines 4-22, which shows that the debugger program is stored in a ROM accessible to the processor).

With respect to claim 12 (original), the rejection of claim 10 is incorporated, and Marik in view of Fritz further teaches or suggests at least one symbol file stored in at least one of the memory and a storage device accessible by the processor (see, for example, column 8, lines 4-22, which shows that the program under test and implicitly a corresponding symbol file is stored in a RAM accessible to the processor).

With respect to claim 13 (original), the rejection of claim 10 is incorporated, and Marik in view of Fritz further teaches or suggests:

a test script handler, wherein at least a portion of the test script handler is encoded as instructions stored in the memory and executable on the processor (see, for example, FIG. 5 of Fritz, which shows such a test script handler in the form of script interpreter 107).

With respect to claim 14 (original), the rejection of claim 13 is incorporated, and Marik in view of Fritz further teaches or suggests that the test script handler is further configured to send the plurality of test commands to the debugger agent (see, for example, Fritz, column 4, lines 27-40, which shows that the test commands are sent from the test script handler).

With respect to claim 15 (original), the rejection of claim 10 is incorporated, and Marik in view of Fritz further teaches or suggests:

a second memory (see, for example, hardware debug device 101 in FIG. 6 of Fritz, which implicitly includes a second memory);

a second processor coupled to the second memory (see, for example, hardware debug device 101 in FIG. 6 of Fritz, which includes a second processor 204 implicitly coupled to the second memory); and

a test script handler, wherein at least a portion of the test script handler is encoded as instructions stored in the second memory and executable on the second processor (see, for example, FIG. 5 of Fritz, which shows that hardware debug device 101 includes such a test script handler in the form of script interpreter 107).

With respect to claim 16 (original), the rejection of claim 15 is incorporated, and Marik in view of Fritz further teaches or suggests that the test script handler is further configured to

send the plurality of test commands to the debugger agent (see, for example, Fritz, column 4, lines 27-40, which shows that the test commands are sent from the test script handler).

With respect to claims 17-19 (original) and 20-24 (previously presented), the limitations recited in the claims are analogous to those of claims 2-9, respectively (see the rejection of claims 2-9 above).

With respect to claim 25 (currently amended), the claim is directed to a computer readable storage medium that corresponds to the method recited in claim 1 (see the rejection of claim 1 above).

With respect to claim 26-33 (previously presented), the limitations recited in the claims are analogous to those of claims 2-9, respectively (see the rejection of claims 2-9 above).

With respect to claim 34 (currently amended), Marik teaches an apparatus comprising:
a means for selecting a debugger program suitable for a device under test (see, for example, column 2, lines 23-31, which shows a debugger program suitable for a device under test, and note that the debugger agent is implicitly configured to select the debugger program).

Marik does not explicitly describe that the means for selecting is configured to select the debugger program from a plurality of debugger programs.

Nonetheless, in an analogous art, Akgul teaches a method and system for debugging a device under test (see, for example, FIG. 1 and paragraph [0017]). The system selects a debugger module from a plurality of debugger modules and loads the debugger module into the device under test (see, for example, paragraph [0042]). Akgul describes that loading only the

necessary debugger modules conserves storage space on the device under test (see, for example, paragraph [0018]).

One of ordinary skill in the art could, with predictable results, incorporate such selection and loading into the teachings of Marik. Thus, it would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the teachings of Marik such that the means for selecting is configured to select the debugger program from a plurality of debugger programs. As Akgul suggests, such an implementation would conserve storage space on the device under test.

Marik in view of Akgul further teaches or suggests:

a means for causing the debugger program to be loaded into a device under test, wherein the device under test is configured to execute the program under test (see, for example, column 8, lines 23-25, which shows loading a debugger program, and column 2, lines 23-31, which shows that the debugger program is loaded into a device under test that is configured to execute a program under test, and see, for example, column 12, lines 13-18, which further shows that debugger routines are downloaded into the device under test);

a means for sending a plurality of test commands to the device under test (see, for example, column 8, line 53 to column 9, line 1, which shows sending a plurality of test commands in the form of D-packets to the device under test).

Marik does not explicitly describe that the plurality of test commands are sent to the device under test according to a test script.

Nonetheless, in an analogous art, Fritz teaches a method and system for debugging a device under test (see, for example, column 4, lines 13-21). Specifically, Fritz teaches sending a

plurality of test commands to the device under test according to a test script (see, for example, column 4, lines 27-40). The test script enables the user to define complex test actions (see, for example, column 6, lines 29-55).

One of ordinary skill in the art could, with predictable results, incorporate such a test script into the teachings of Marik. Thus, it would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the teachings of Marik such that the plurality of test commands are sent to the device under test according to a test script. As Fritz suggests, such an implementation would enable the user to define complex test actions.

Marik in view of Akgul and Fritz further teaches or suggests:

a means for activating the means for debugging when a watched event occurs during execution of the program under test (see, for example, column 8, lines 34-50, which shows activating the debugger program when a watched event is encountered in the program under test).

With respect to claims 35, 36 (original) and 37-39 (previously presented), the limitations recited in the claims are analogous to those of claims 2, 3 and 5-7, respectively (see the rejection of claims 2, 3 and 5-7 above).

Conclusion

7. The prior art made of record and not relied upon is considered pertinent to Applicant's disclosure (see the attached Notice of References of Cited).

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is 571-272-3707.

The examiner can normally be reached on Monday to Friday from 8:00 AM to 4:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Michael J. Yigdall
Examiner
Art Unit 2192

/Michael J. Yigdall/
Examiner, Art Unit 2192